Class_10

Isabella Ruud PID: 59016138

Table of contents

THe PDB database

The main repository for biomolecular data is called the PDB (protein data bank) and can be found at: https://www.rcsb.org/

Let's see what it contains in terms of molecule and method of structure determination (Analyze > PDB stats > By Mol type and method)

```
pdbstats <- read.csv("Data Export Summary.csv")
head(pdbstats)
```

```
           Molecular.Type   X.ray     EM    NMR Multiple.methods Neutron Other
1          Protein (only) 169,563 16,774 12,578              208      81    32
2 Protein/Oligosaccharide   9,939  2,839     34                8       2     0
3              Protein/NA   8,801  5,062    286                7       0     0
4     Nucleic acid (only)   2,890    151  1,521               14       3     1
5                   Other     170     10     33                0       0     0
6  Oligosaccharide (only)      11      0      6                1       0     4
    Total
1 199,236
2  12,822
3  14,156
```

```
4   4,580
5     213
6      22
```

Need to get rid of the commas using sub in the numbers and then convert the chars to ints using as.numeric

```
as.numeric(sub(",", "", pdbstats$X.ray))
```

```
[1] 169563   9939   8801   2890    170     11
```

Or can use readr package instead

```
library(readr)
pdbstats <- read_csv("Data Export Summary.csv")
```

```
Rows: 6 Columns: 8
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): Molecular Type
dbl (3): Multiple methods, Neutron, Other
num (4): X-ray, EM, NMR, Total

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(pdbstats)
```

```
# A tibble: 6 x 8
  `Molecular Type`   `X-ray`    EM   NMR `Multiple methods` Neutron Other  Total
  <chr>                <dbl> <dbl> <dbl>              <dbl>   <dbl> <dbl>  <dbl>
1 Protein (only)      169563 16774 12578                208      81    32 199236
2 Protein/Oligosacc~    9939  2839    34                  8       2     0  12822
3 Protein/NA            8801  5062   286                  7       0     0  14156
4 Nucleic acid (onl~    2890   151  1521                 14       3     1   4580
5 Other                  170    10    33                  0       0     0    213
6 Oligosaccharide (~      11     0     6                  1       0     4     22
```

Now need to rename the column names so that they do not have spaces or mixes up upper/lower case. Can use janitor package for this and its clean_names() function for this

```
library(janitor)
```

```
Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```
colnames(pdbstats)
```

```
[1] "Molecular Type"   "X-ray"            "EM"               "NMR"
[5] "Multiple methods" "Neutron"          "Other"            "Total"
```

```
pdbstats <- clean_names(pdbstats)
```

Q1: What percentage of structures in the PDB are solved by X-Ray and Electron Microscopy.

```
xray <- sum(pdbstats$x_ray)
EM <- sum(pdbstats$em)
total <- sum(pdbstats$total)

percent_xray <- xray / total
percent_EM <- EM / total

paste("percent x ray: ", percent_xray)
```

```
[1] "percent x ray:  0.828354881854659"
```

```
paste("percent EM: ", percent_EM)
```

```
[1] "percent EM:  0.107501655636305"
```

82.8% of structures in the PDB are solved by X-ray 10.7% of structures are solved by electron miscroscopy

Q2: What proportion of structures in the PDB are protein?

```
round(pdbstats$total[1]/total * 100, digits=2)
```

```
[1] 86.24
```

86.24% of structurs in the PDB are protein

There are 253,206,171 proteins in UniProt and there are only 231,029 known structures in in the PDB. this is a tiny fraction!

```
total/253206171
```

```
[1] 0.0009124146
```

In the next lab we will use prediction methods that approach the accuracy of xray crystallography.

Molecular visualization with Mol*

> Q3: Type HIV in the PDB website search box on the home page and determine how many HIV-1 protease structures are in the current PDB?

There are 5 HIV-1 protease structures

Mol-star is a new online structure viewer that is taking over the world of biomolecular visualization. let's see how to use it from https://molstar.org/

My first image from Mol* of 1HSG

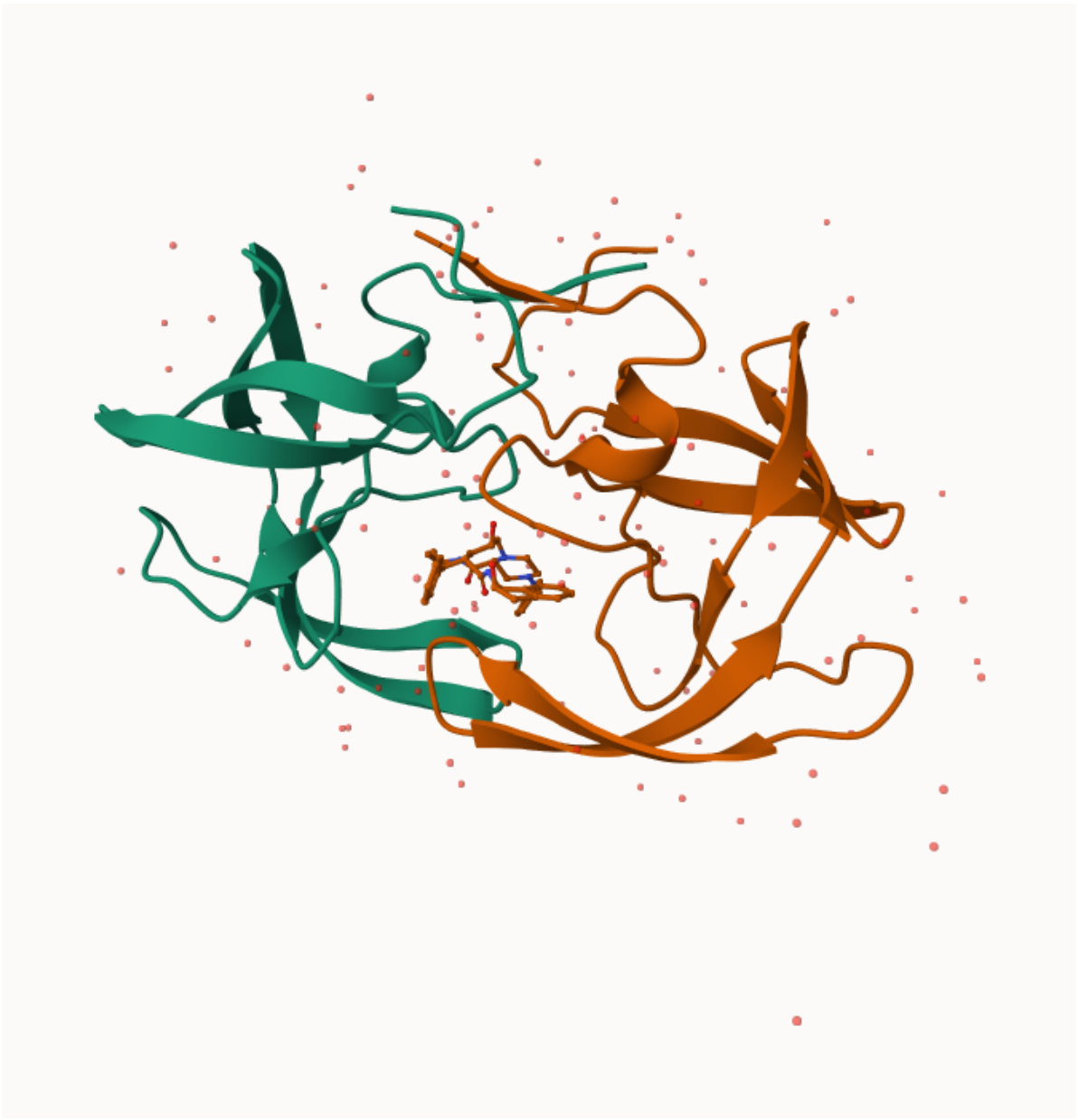


Figure 1: Fig 1. A first view of the HIV-pr dimer

Q4: Water molecules normally have 3 atoms. Why do we see just one atom per water molecule in this structure?

The structure was solved with a resolution of 2 angstrom, and since hydrogen atoms are so small, they are not seen in the structure even though they are actually supposed to be there

> Q5: There is a critical "conserved" water molecule in the binding site. Can you identify this water molecule? What residue number does this water molecule have

Yes, it forms 2 hydrogren bonds with the ligand and 2 hydrogen bonds with the protease. It is called HOH308

> Q6: Generate and save a figure clearly showing the two distinct chains of HIV-protease along with the ligand. You might also consider showing the catalytic residues ASP 25 in each chain and the critical water (we recommend "Ball & Stick" for these side-chains). Add this figure to your Quarto document.

I want an image of the binding cleft for the MK1 inhibitor, and image of the most valubale water in human history, and an image showing the catalytic ASP amino acid.

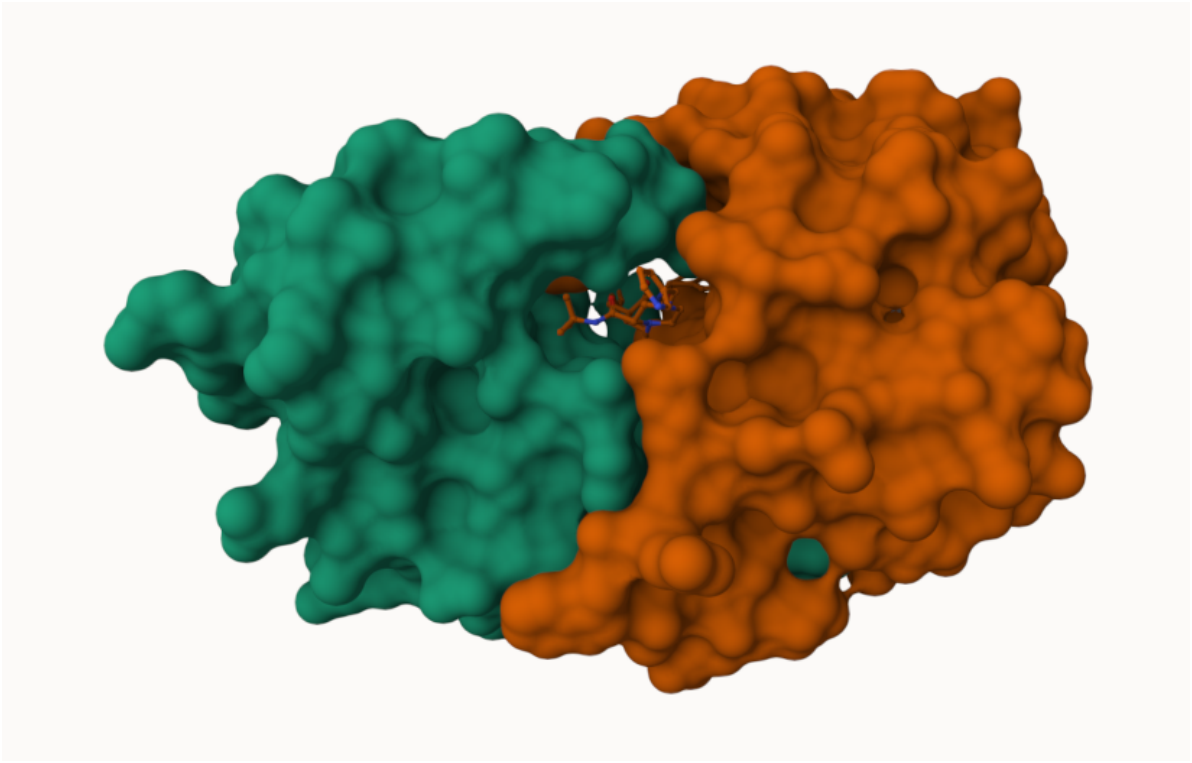


Figure 2: Fig 2a. Substrate binding cleft

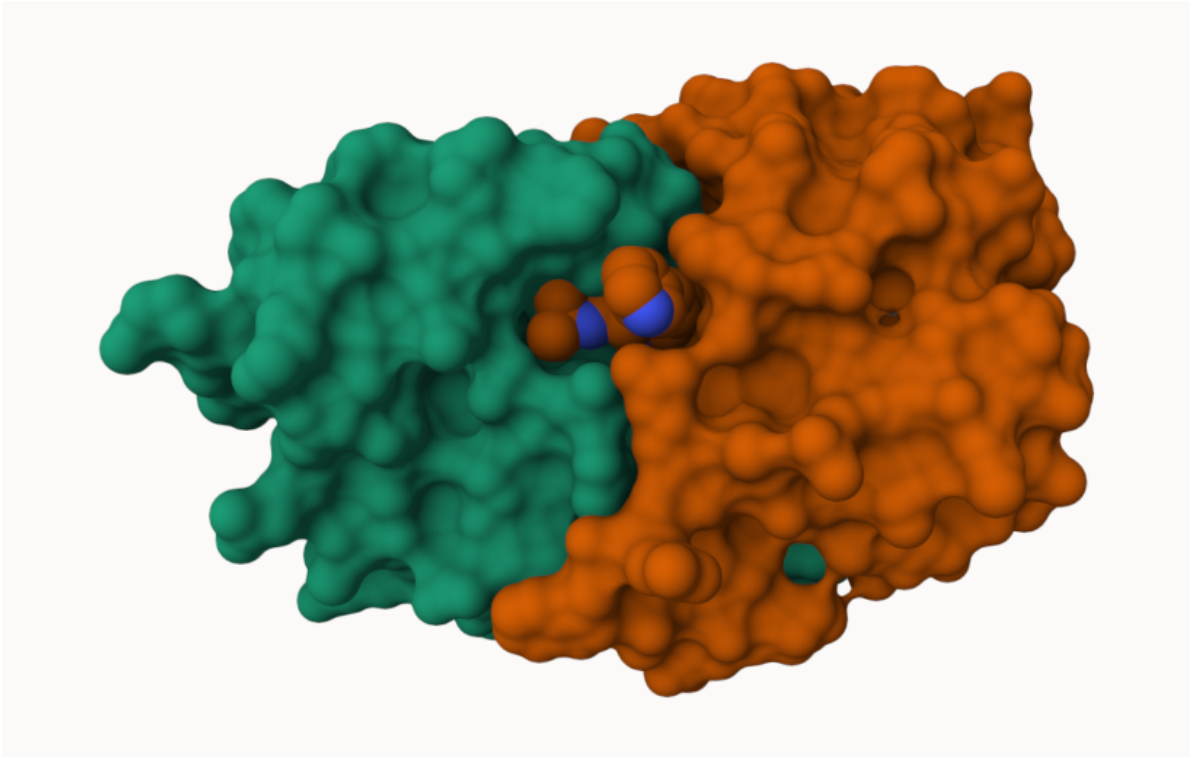


Figure 3: Fig 2b. Substrate binding cleft, ligand space fill

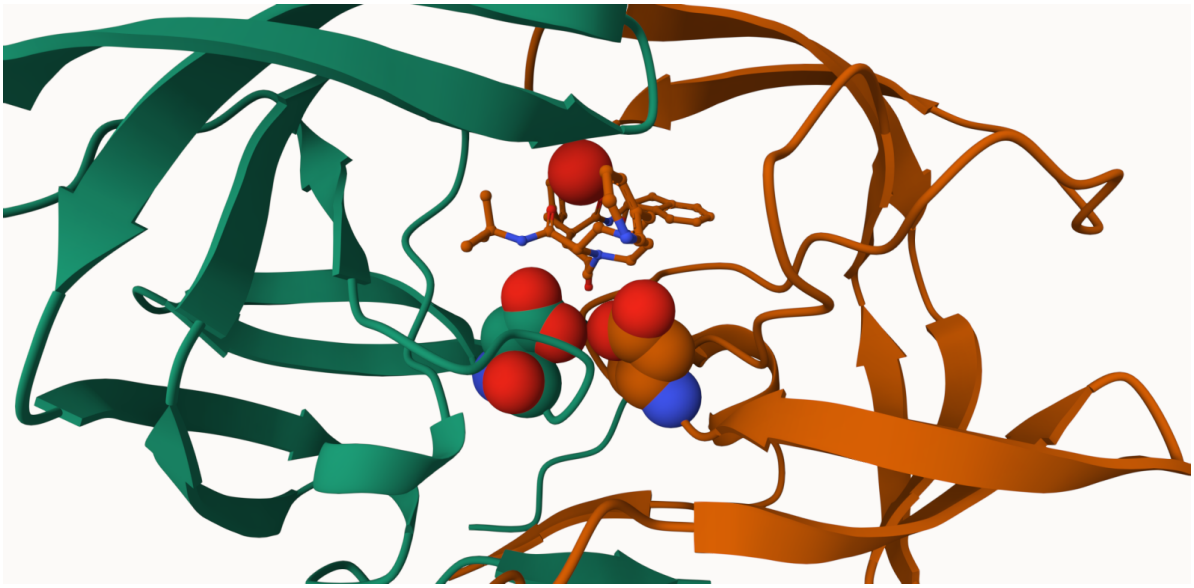


Figure 4: Fig 3. Important water molecule and catalytic aspartic acids

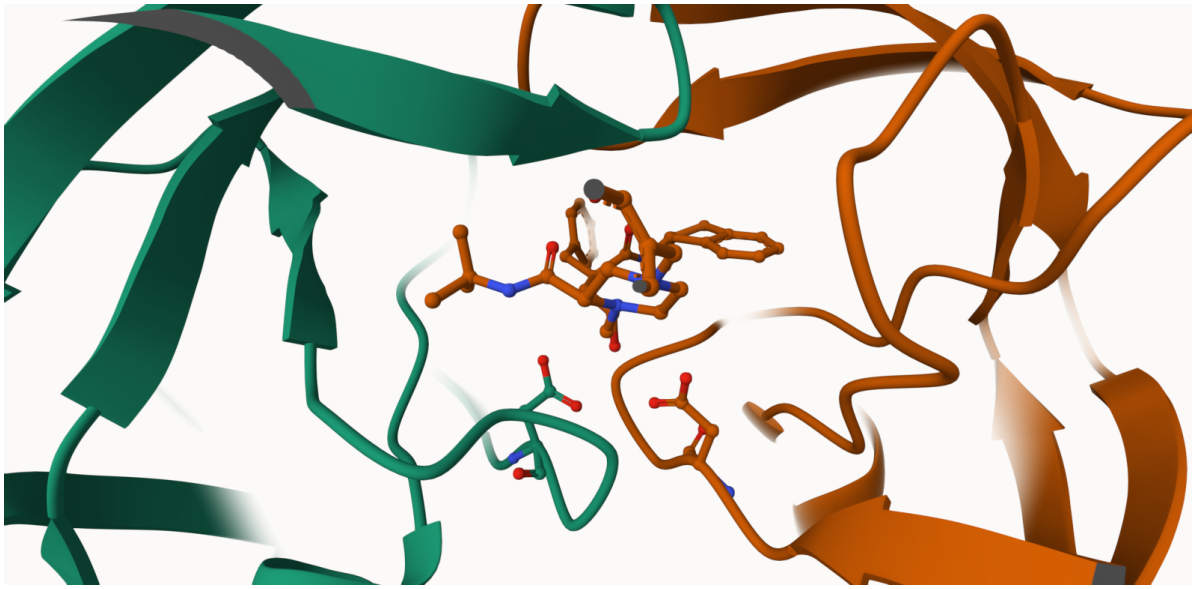


Figure 5: Fig 3a. Important water molecule and catalytic aspartic acids, ball and stick

Section 3: Using the Bio3d package

This package has tons of tools and utilities for structural bioinformatics.

Can read in from the online databank if you give it an accession number

```
library(bio3d)
hiv <- read.pdb("1hsg")
```

```
  Note: Accessing on-line PDB file
```

```
hiv
```

```
 Call:  read.pdb(file = "1hsg")

   Total Models#: 1
     Total Atoms#: 1686,  XYZs#: 5058  Chains#: 2  (values: A B)

     Protein Atoms#: 1514  (residues/Calpha atoms#: 198)
     Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)
```

```
     Non-protein/nucleic Atoms#: 172  (residues: 128)
     Non-protein/nucleic resid values: [ HOH (127), MK1 (1) ]

   Protein sequence:
      PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYD
      QILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFPQITLWQRPLVTIKIGGQLKE
      ALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYDQILIEICGHKAIGTVLVGPTP
      VNIIGRNLLTQIGCTLNF

+ attr: atom, xyz, seqres, helix, sheet,
        calpha, remark, call
```

Q7: How many amino acid residues are there in this pdb object?

198 amino acid residues in the pbd object

Can easily get the sequence out. How long is this sequence/how many amino acids are in the structure?

```
s <- pdbseq(hiv)
length(s)
```

```
[1] 198
```

The sequence is 198 amino acids long

Q8: Name one of the two non-protein residues?

MK1 is one of the two non-protein residues

Q9: How many protein chains are in this structure?

There are 2 protein chains

```
head(hiv$atom)
```

```
  type eleno elety  alt resid chain resno insert      x      y     z o     b
1 ATOM     1     N <NA>   PRO     A     1   <NA> 29.361 39.686 5.862 1 38.10
2 ATOM     2    CA <NA>   PRO     A     1   <NA> 30.307 38.663 5.319 1 40.62
3 ATOM     3     C <NA>   PRO     A     1   <NA> 29.760 38.071 4.022 1 42.64
4 ATOM     4     O <NA>   PRO     A     1   <NA> 28.600 38.302 3.676 1 43.40
5 ATOM     5    CB <NA>   PRO     A     1   <NA> 30.508 37.541 6.342 1 37.87
6 ATOM     6    CG <NA>   PRO     A     1   <NA> 29.296 37.591 7.162 1 38.40
```

```
  segid elesy charge
1  <NA>     N   <NA>
2  <NA>     C   <NA>
3  <NA>     C   <NA>
4  <NA>     O   <NA>
5  <NA>     C   <NA>
6  <NA>     C   <NA>
```

Predict the functional motions

Let's read a new structure "6s36"

```
pdb <- read.pdb("6s36")
```

```
  Note: Accessing on-line PDB file
   PDB has ALT records, taking A only, rm.alt=TRUE
```

```
pdb
```

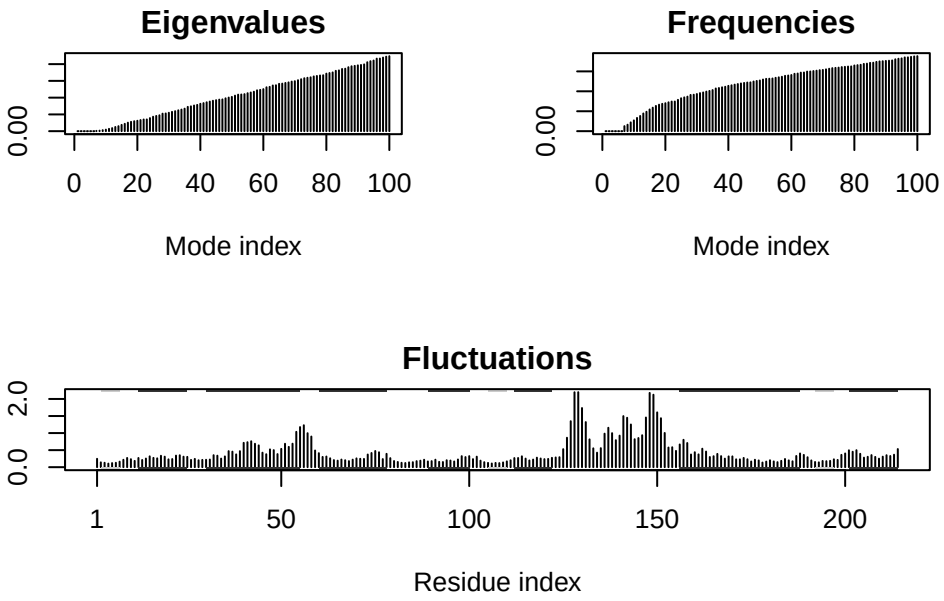
```
 Call:  read.pdb(file = "6s36")

   Total Models#: 1
     Total Atoms#: 1898,  XYZs#: 5694  Chains#: 1  (values: A)

     Protein Atoms#: 1654  (residues/Calpha atoms#: 214)
     Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)

     Non-protein/nucleic Atoms#: 244  (residues: 244)
     Non-protein/nucleic resid values: [ CL (3), HOH (238), MG (2), NA (1) ]

   Protein sequence:
      MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
      DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDKI
      VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
      YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG

+ attr: atom, xyz, seqres, helix, sheet,
        calpha, remark, call
```

We can run a NMA calculation on this structure:

```
m <- nma(pdb)
```

```
 Building Hessian...        Done in 0.03 seconds.
 Diagonalizing Hessian...   Done in 0.28 seconds.
```

```
plot(m, sse=pdb)
```



We can write out a trajectory of the predicted dynamics using mktrj() function

```
mktrj(m, file="results.pdb")
```

Section 4: Comparative analysis

Q10. Which of the packages above is found only on BioConductor and not CRAN?

the msa package

Q11. Which of the above packages is not found on BioConductor or CRAN?

the bio3d-view package

Q12. True or False? Functions from the devtools package can be used to install packages from GitHub and BitBucket?

True

```
aa <- get.seq("1ake_A")
```

```
Warning in get.seq("1ake_A"): Removing existing file: seqs.fasta
```

```
Fetching... Please wait. Done.
```

```
aa
```

```
              1        .         .         .         .         .         60
pdb|1AKE|A    MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
              1        .         .         .         .         .         60

             61        .         .         .         .         .         120
pdb|1AKE|A    DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
             61        .         .         .         .         .         120

            121        .         .         .         .         .         180
pdb|1AKE|A    VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
            121        .         .         .         .         .         180

            181        .         .         .   214
pdb|1AKE|A    YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG
            181        .         .         .   214

Call:
  read.fasta(file = outfile)

Class:
  fasta

Alignment dimensions:
  1 sequence rows; 214 position columns (214 non-gap, 0 gap)

+ attr: id, ali, call
```

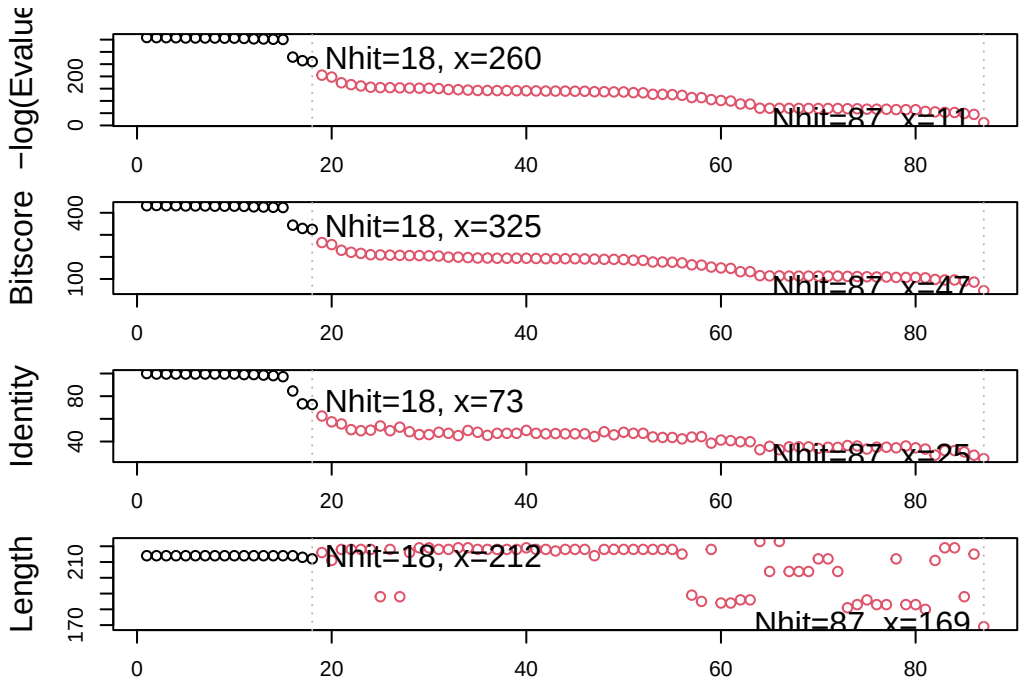
Search the PDB database for related sequences

```
blast <- blast.pdb(aa)
```

```
 Searching ... please wait (updates every 5 seconds) RID = UD7R2Y1M013
 .........................
 Reporting 87 hits
```

```
hits <- plot(blast)
```

```
  * Possible cutoff values:    260 11
            Yielding Nhits:    18 87

  * Chosen cutoff value of:    260
            Yielding Nhits:    18
```



see the hits from the blast

```
hits
```

```
$hits
   pdb.id   acc      group
1  "1AKE_A" "1AKE_A" "1"
```

```
2  "8BQF_A" "8BQF_A" "1"
3  "4X8M_A" "4X8M_A" "1"
4  "6S36_A" "6S36_A" "1"
5  "8Q2B_A" "8Q2B_A" "1"
6  "8RJ9_A" "8RJ9_A" "1"
7  "6RZE_A" "6RZE_A" "1"
8  "4X8H_A" "4X8H_A" "1"
9  "3HPR_A" "3HPR_A" "1"
10 "1E4V_A" "1E4V_A" "1"
11 "5EJE_A" "5EJE_A" "1"
12 "1E4Y_A" "1E4Y_A" "1"
13 "3X2S_A" "3X2S_A" "1"
14 "6HAP_A" "6HAP_A" "1"
15 "6HAM_A" "6HAM_A" "1"
16 "8PVW_A" "8PVW_A" "1"
17 "4K46_A" "4K46_A" "1"
18 "4NP6_A" "4NP6_A" "1"

$pdb.id
 [1] "1AKE_A" "8BQF_A" "4X8M_A" "6S36_A" "8Q2B_A" "8RJ9_A" "6RZE_A" "4X8H_A"
 [9] "3HPR_A" "1E4V_A" "5EJE_A" "1E4Y_A" "3X2S_A" "6HAP_A" "6HAM_A" "8PVW_A"
[17] "4K46_A" "4NP6_A"

$acc
 [1] "1AKE_A" "8BQF_A" "4X8M_A" "6S36_A" "8Q2B_A" "8RJ9_A" "6RZE_A" "4X8H_A"
 [9] "3HPR_A" "1E4V_A" "5EJE_A" "1E4Y_A" "3X2S_A" "6HAP_A" "6HAM_A" "8PVW_A"
[17] "4K46_A" "4NP6_A"

$inds
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
[25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[85] FALSE FALSE FALSE

attr(,"class")
[1] "blast"
```

```
head(blast$raw)
```

```
        queryid subjectids identity alignmentlength mismatches gapopens q.start
1 Query_7605673     1AKE_A  100.000             214          0        0       1
2 Query_7605673     8BQF_A   99.533             214          1        0       1
3 Query_7605673     4X8M_A   99.533             214          1        0       1
4 Query_7605673     6S36_A   99.533             214          1        0       1
5 Query_7605673     8Q2B_A   99.533             214          1        0       1
6 Query_7605673     8RJ9_A   99.533             214          1        0       1
  q.end s.start s.end    evalue bitscore positives
1   214       1   214 1.61e-156      432    100.00
2   214      21   234 2.64e-156      433    100.00
3   214       1   214 2.89e-156      432    100.00
4   214       1   214 4.24e-156      432    100.00
5   214       1   214 1.13e-155      431     99.53
6   214       1   214 1.13e-155      431     99.53
```

download all these structures to our project directory

```
hits$pdb.id
```

```
 [1] "1AKE_A" "8BQF_A" "4X8M_A" "6S36_A" "8Q2B_A" "8RJ9_A" "6RZE_A" "4X8H_A"
 [9] "3HPR_A" "1E4V_A" "5EJE_A" "1E4Y_A" "3X2S_A" "6HAP_A" "6HAM_A" "8PVW_A"
[17] "4K46_A" "4NP6_A"
```

```
files <- get.pdb(hits$pdb.id, path="pdbs", split=TRUE, gzip=TRUE)
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1AKE.pdb exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8BQF.pdb exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4X8M.pdb exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6S36.pdb exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8Q2B.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8RJ9.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6RZE.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4X8H.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/3HPR.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1E4V.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/5EJE.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1E4Y.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/3X2S.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6HAP.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6HAM.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8PVW.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4K46.pdb exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4NP6.pdb exists. Skipping download
```

```
  |
  |                                                                      |   0%
  |
  |====                                                                  |   6%
  |
  |========                                                              |  11%
  |
  |============                                                          |  17%
  |
  |================                                                      |  22%
  |
  |====================                                                  |  28%
  |
  |========================                                              |  33%
  |
  |============================                                          |  39%
  |
  |================================                                      |  44%
  |
  |====================================                                  |  50%
  |
  |========================================                              |  56%
  |
  |============================================                          |  61%
  |
  |================================================                      |  67%
  |
  |====================================================                  |  72%
  |
  |========================================================              |  78%
  |
  |============================================================          |  83%
  |
  |================================================================      |  89%
  |
  |====================================================================  |  94%
  |
  |======================================================================| 100%
```

Align and supperpose

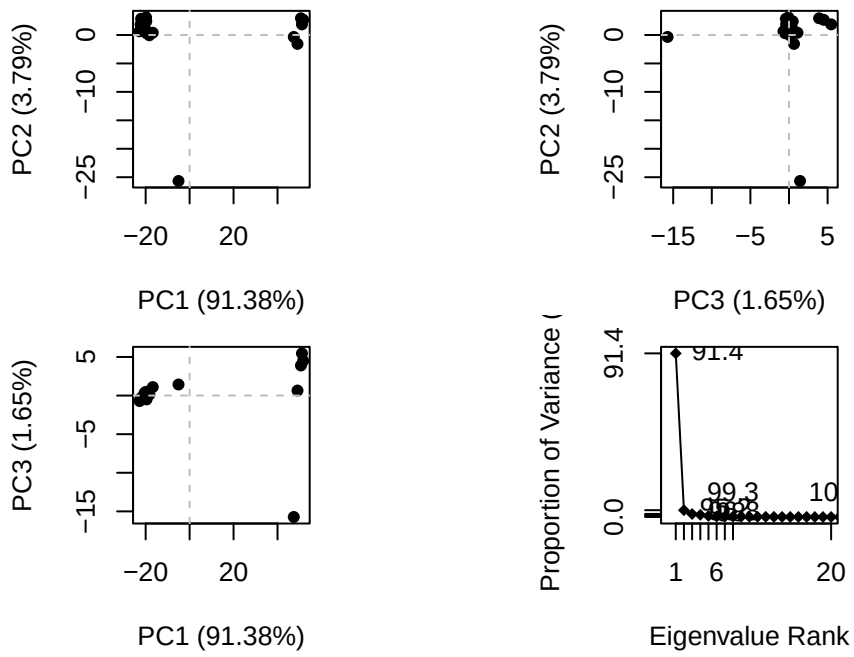
```
pdbs <- pdbaln(files, fit = TRUE, exefile="msa")
```

```
Reading PDB files:
pdbs/split_chain/1AKE_A.pdb
pdbs/split_chain/8BQF_A.pdb
pdbs/split_chain/4X8M_A.pdb
pdbs/split_chain/6S36_A.pdb
pdbs/split_chain/8Q2B_A.pdb
pdbs/split_chain/8RJ9_A.pdb
pdbs/split_chain/6RZE_A.pdb
pdbs/split_chain/4X8H_A.pdb
pdbs/split_chain/3HPR_A.pdb
pdbs/split_chain/1E4V_A.pdb
pdbs/split_chain/5EJE_A.pdb
pdbs/split_chain/1E4Y_A.pdb
pdbs/split_chain/3X2S_A.pdb
pdbs/split_chain/6HAP_A.pdb
pdbs/split_chain/6HAM_A.pdb
pdbs/split_chain/8PVW_A.pdb
pdbs/split_chain/4K46_A.pdb
pdbs/split_chain/4NP6_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
..   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
..   PDB has ALT records, taking A only, rm.alt=TRUE
..   PDB has ALT records, taking A only, rm.alt=TRUE
....   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
..

Extracting sequences

pdb/seq: 1   name: pdbs/split_chain/1AKE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 2   name: pdbs/split_chain/8BQF_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
```

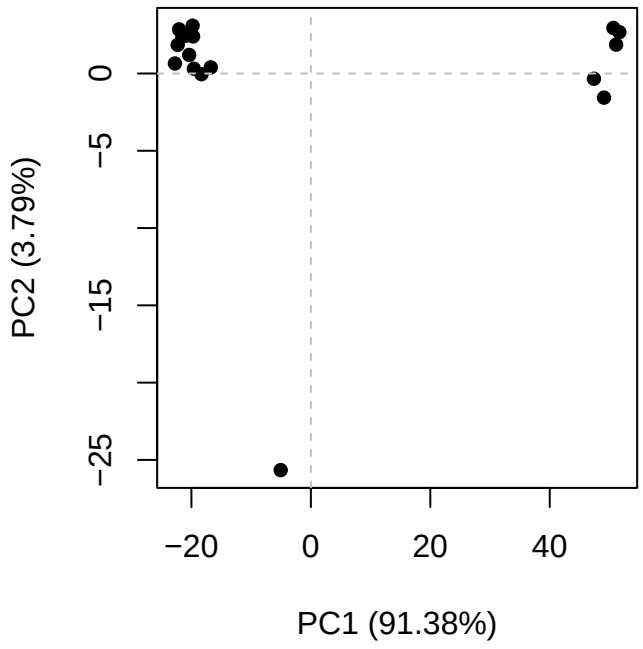
```
pdb/seq: 3   name: pdbs/split_chain/4X8M_A.pdb
pdb/seq: 4   name: pdbs/split_chain/6S36_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 5   name: pdbs/split_chain/8Q2B_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 6   name: pdbs/split_chain/8RJ9_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 7   name: pdbs/split_chain/6RZE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 8   name: pdbs/split_chain/4X8H_A.pdb
pdb/seq: 9   name: pdbs/split_chain/3HPR_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 10   name: pdbs/split_chain/1E4V_A.pdb
pdb/seq: 11   name: pdbs/split_chain/5EJE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 12   name: pdbs/split_chain/1E4Y_A.pdb
pdb/seq: 13   name: pdbs/split_chain/3X2S_A.pdb
pdb/seq: 14   name: pdbs/split_chain/6HAP_A.pdb
pdb/seq: 15   name: pdbs/split_chain/6HAM_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 16   name: pdbs/split_chain/8PVW_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 17   name: pdbs/split_chain/4K46_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 18   name: pdbs/split_chain/4NP6_A.pdb
```

PCA analysis of the aligned structures

```
pc.xray <- pca(pdbs)
plot(pc.xray)
```




```
plot(pc.xray, pc.axes = c(1,2))
```



We can view the main PC1 cpatured displacements with the mktrj() function

```
pc1 <- mktrj(pc.xray, pc=1, file="pc_1.pdb")
```

```
modes <- nma(pdbs)
```

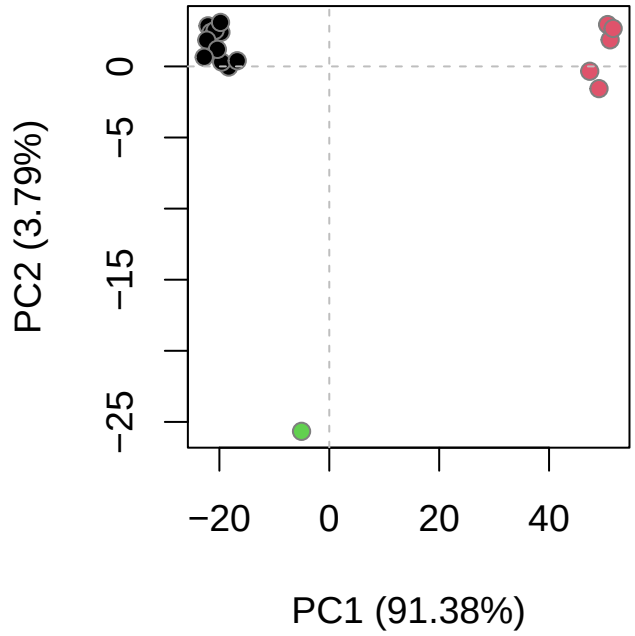
```
Warning in nma.pdbs(pdbs): 8BQF_A.pdb might have missing residue(s) in structure:
   Fluctuations at neighboring positions may be affected.


Details of Scheduled Calculation:
  ... 18 input structures
  ... storing 540 eigenvectors for each structure
  ... dimension of x$U.subspace: ( 546x540x18 )
  ... coordinate superposition prior to NM calculation
  ... aligned eigenvectors (gap containing positions removed)
  ... estimated memory usage of final 'eNMA' object: 40.6 Mb


  |
  |                                                                      |   0%
  |
  |====                                                                  |   6%
  |
  |========                                                              |  11%
  |
  |============                                                          |  17%
  |
  |================                                                      |  22%
  |
  |====================                                                  |  28%
  |
  |========================                                              |  33%
  |
  |============================                                          |  39%
  |
  |================================                                      |  44%
  |
  |====================================                                  |  50%
  |
  |========================================                              |  56%
  |
  |============================================                          |  61%
```

```
  |
  |=============================================                        |  67%
  |
  |==============================================                       |  72%
  |
  |================================================                     |  78%
  |
  |==================================================                   |  83%
  |
  |====================================================                 |  89%
  |
  |======================================================               |  94%
  |
  |========================================================================| 100%
```

```
rd <- rmsd(pdbs)
```

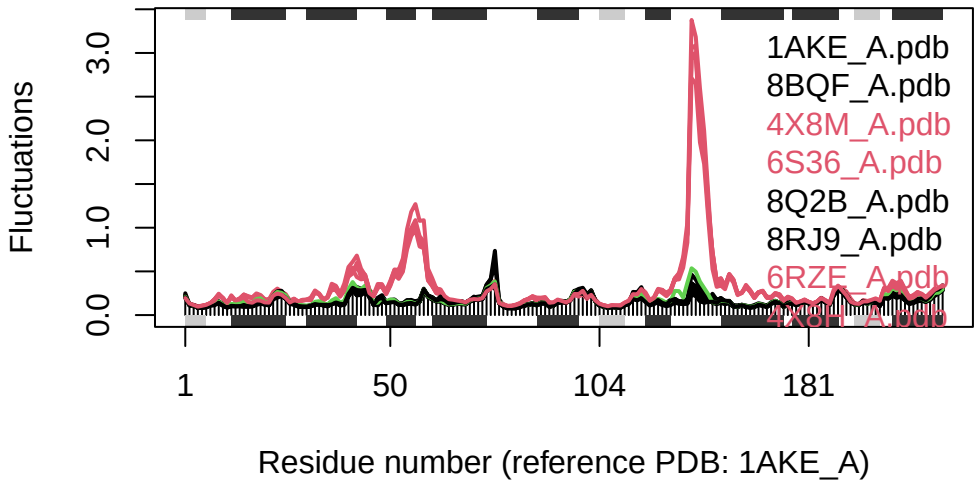
```
Warning in rmsd(pdbs): No indices provided, using the 182 non NA positions
```

```
# Structure-based clustering
hc.rd <- hclust(dist(rd))
grps.rd <- cutree(hc.rd, k=3)

plot(pc.xray, 1:2, col="grey50", bg=grps.rd, pch=21, cex=1)
```




```
plot(modes, pdbs, col=grps.rd)
```

```
Extracting SSE from pdbs$sse attribute
```



Q14. What do you note about this plot? Are the black and colored lines similar or different? Where do you think they differ most and why?

The black and colored lines are similar in some areas but very different in others. The black and colored lines are the most similar over the alpha helix and beta sheet secondary structure regions (black and gray bars) and they are the most different over the loop regions (white spaces between bars).